

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
APPLICATION FOR LETTERS PATENT

**Verifying the Presence of an Original  
Data Storage Medium**

Inventor:  
Paul England

ATTORNEY'S DOCKET NO. MS1-408US

1      **TECHNICAL FIELD**

2      This invention relates to verification systems and methods. More  
3      particularly, the invention relates to systems and methods that verify the existence  
4      of an original data storage medium, such as a compact disc.

5

6      **BACKGROUND OF THE INVENTION**

7      Application programs for use on computer-based systems are often  
8      distributed on compact discs (CDs) or digital versatile discs (DVDs). DVDs may  
9      also be referred to as digital video discs. Generally, the content of entire CDs and  
10     DVDs is too large to distribute across the Internet (CDs can store more than 500  
11     MB of data and DVDs are capable of storing more than 4 GB of data). However,  
12     many application programs that are distributed on CD or DVD utilize only a small  
13     portion of the available storage space on the CD or DVD. If the size of the  
14     application program is small enough, pirated copies of the software might be  
15     distributed across a network, such as the Internet. To prevent this type of software  
16     piracy, it is desirable to determine whether an original CD or DVD (containing the  
17     application program) is present in a computing device that is attempting to launch  
18     the application.

19     To determine whether an original CD or DVD is present, an undesirable  
20     solution provides a verification system that uses a computer system to compare  
21     every byte of data stored on the CD or DVD with a known valid copy of the data  
22     stored on the computer system. This solution is undesirable for two reasons.  
23     First, the solution requires reading the entire CD or DVD, which is time-  
24     consuming. Second, the solution requires storage of a known valid copy of the  
25

1 data on the computer system. Storage of such a large amount of data may not be  
2 practical on many computer systems. Further, the required storage space is  
3 increased with each new application program that must be verified by the  
4 computer system.

5 Another problem arises when software pirates combine multiple application  
6 programs on a single CD or DVD. Since many application programs do not use  
7 the entire storage space available on the CD or DVD, software pirates create CDs  
8 or DVDs that contain multiple application programs. Although the actual program  
9 code may be identical to a legitimate copy of the application program, the excess  
10 data stored on the pirated CD or DVD does not match the corresponding lack of  
11 data on the legitimate CD or DVD. Thus, it is desirable to provide a verification  
12 system that is able to identify otherwise accurate copies of application programs  
13 improperly stored on a CD or DVD with other application programs.

14 Similarly, music is commonly distributed on CDs and, to a lesser degree,  
15 on DVDs. The manufacturers of certain music CDs and DVDs may offer  
16 additional products or services to customers who purchase music CDs and DVDs.  
17 Before offering these additional products or services, the manufacturer must verify  
18 that the individual requesting the product or service has obtained a legitimate copy  
19 of the original music CD or DVD.

20 Furthermore, customers purchasing legitimate music CDs and DVDs may  
21 use an application program (commonly referred to as a “ripper” application) to  
22 extract raw audio data from a CD or DVD and convert the raw audio data to a  
23 particular format, such as MP3 (MPEG Audio Layer 3). MP3 is an audio  
24 compression technology that compresses CD-quality audio data into music files.  
25

1 MP3 music files are played back on a computer system using an appropriate  
2 software program or installed, for example, on a handheld device for playback. To  
3 prevent unauthorized copying or distribution of MP3 music files, it is desirable to  
4 verify that the user attempting to play or install an MP3 music file has a legitimate  
5 copy of the original music CD or DVD.

6 One solution to this verification problem is to read a particular piece of data  
7 from the CD or DVD, such as the volume identifier. The verification system  
8 compares the volume identifier read from the CD or DVD to an expected value. If  
9 the volume identifier matches the expected value, then the CD or DVD is  
10 "verified." This verification solution is easily defeated by copying the particular  
11 piece of data to the appropriate location on the pirated CD or DVD. Thus, a better  
12 verification solution is needed to discourage piracy.

13 As discussed above, attempting to compare every byte of data stored on a  
14 CD or DVD with a known valid copy of the data is impractical. The present  
15 invention provides a system that verifies the existence of an original data storage  
16 medium, such as a CD or DVD, without requiring an analysis of every byte of data  
17 stored on the CD or DVD.

18

19 **SUMMARY OF THE INVENTION**

20 The present invention allows a computer system or other device to  
21 determine whether an original CD or DVD is present. If an original CD or DVD  
22 is not present, the requested application will not launch or the requested music will  
23 not play. Thus, pirated media or pirated software distributed without an original  
24 CD or DVD will not function properly. The invention determines whether an  
25

1 original CD or DVD is present by requesting one or more randomly chosen data  
2 blocks from the CD or DVD. The requested data is read from the CD or DVD and  
3 verified against known valid data (e.g., the data that is present on a legitimate CD  
4 or DVD). If the requested data matches the known valid data, then the requested  
5 operation (e.g., launch an application program or play a music file) is allowed.  
6 Since the requested data segments are chosen at random, a pirate cannot know  
7 which data segments will be chosen for verification. Furthermore, different data  
8 segments are selected during each verification process.

9 Particular embodiments of the invention partition the removable data  
10 storage medium into multiple blocks of data. A cryptographic digest is then  
11 calculated for each data block. The digests are compared to determine whether the  
12 retrieved data matches the verification data.

13 In one embodiment of the invention the removable data storage medium is  
14 a compact disc (CD).

15 In another embodiment of the invention, the removable data storage  
16 medium is a digital versatile disc (DVD).

17 An implementation of the invention provides a verification system  
18 including a data reading device that reads data from a removable data storage  
19 medium. A verification module coupled to the data reading device randomly  
20 retrieves data from the removable data storage medium. The verification module  
21 compares the retrieved data to corresponding verification data that is known to be  
22 valid. The verification module determines that a legitimate removable data  
23 storage medium is present if the retrieved data matches the corresponding  
24 verification data.

25

1

2 **BRIEF DESCRIPTION OF THE DRAWINGS**

3 Fig. 1 illustrates an exemplary network environment in which a personal  
4 computer is able to retrieve music files across the Internet and install the music  
5 files in a handheld music player.

6 Fig. 2 is a flow diagram illustrating a procedure for verifying the existence  
7 of an original data storage medium before playing or installing a music file.

8 Fig. 3 illustrates a table of digests for a particular music CD.

9 Fig. 4 illustrates an exemplary computer system containing two application  
10 programs and a verification module.

11 Fig. 5 is a flow diagram illustrating a procedure for verifying the existence  
12 of an original compact disc before launching an application program.

13 Fig. 6 illustrates an exemplary environment in which a remote server  
14 verifies whether an original CD is present in a local computer system.

15 Fig. 7 is a block diagram showing pertinent components of a computer in  
16 accordance with the invention.

17

18 **DETAILED DESCRIPTION**

19 The discussion herein assumes that the reader is familiar with  
20 cryptography. For a basic introduction to cryptography, the reader is directed to a  
21 text written by Bruce Schneier and entitled "Applied Cryptography: Protocols,  
22 Algorithms, and Source Code in C," published by John Wiley & Sons with  
23 copyright 1994 (or second edition with copyright 1996).

24 The present invention provides a verification system that allows a verifying  
25

device to determine whether an original data storage medium (e.g., a compact disc (CD) or digital versatile disc (DVD)) is present. If an original data storage medium is not present, then the requested application or function will not be performed. This verification system effectively disables pirated media or pirated software distributed via a network without an original data storage medium or an exact facsimile. The verification system also disables the operation of pirated application programs stored on a data storage medium along with other pirated application programs. Particular embodiments of the invention are described herein with reference to verifying application programs and audio files, such as music files. However, the teachings of the present invention can be applied to any type of data or data arrangement stored on a data storage medium, and is of particular interest when a functionally equivalent version of the application or passive media (e.g., a song or a video), which is much smaller than the original (through compression, or by omitting parts of the application or data), can be distributed instead of the original.

Fig. 1 illustrates an exemplary network environment in which a personal computer 100 is able to retrieve music files (or other files containing audio data) across the Internet and install the music files in a handheld music player 102. Although the example of Fig. 1 illustrates handheld music player 102, alternate embodiments may utilize any type of audio player or audio playback device. Personal computer 100 and music player 102 are coupled together through a communication link 104. Communication link 104 may utilize any type of communication medium and any communication protocol. In a particular embodiment, communication link 104 is a universal serial bus (USB) connection.

1 Personal computer 100 includes a CD-ROM (compact disc read-only  
2 memory) drive 106. Additionally, personal computer 100 may include a DVD-  
3 ROM (digital versatile disc read-only memory) drive (not shown) instead of or in  
4 addition to CD-ROM drive 106. A CD-ROM and a DVD-ROM are examples of  
5 data reading devices. A DVD-ROM drive may be capable of reading both CDs  
6 and DVDs. Throughout this description of the invention, any reference to a CD or  
7 CD-ROM drive shall be understood to have a similar application to a DVD or a  
8 DVD-ROM drive. For example, any reference to a music file or application  
9 program stored on a CD and read by a CD-ROM drive shall also apply to a music  
10 file or application program stored on a DVD and read by a DVD-ROM drive.  
11 Although the invention is described in examples that include CDs and DVDs, the  
12 teachings of the invention can be applied to any type of removable data storage  
13 medium (such as removable diskettes and removable memory cards).

14 Handheld music player 102 is capable of storing and playing music files  
15 encoded in a format such as MP3 (MPEG Audio Layer 3). Although particular  
16 examples are described herein with reference to MP3, the teachings of the present  
17 invention can be applied to any audio data encoding format. Music player 102  
18 contains a verification module 108 and a table of one or more digests 110.  
19 Verification module 108 verifies that an original music CD is present in CD-ROM  
20 drive 106 and that the CD contains a music file that corresponds to a music file  
21 stored in the music player 102. Digests 110 are cryptographic digests representing  
22 blocks of data on an original CD. As shown in Fig. 1, the verification module 108  
23 and the digests 110 are located in music player 102. However, in alternate  
24 embodiments, the verification module 108 and/or the digests 110 may be located  
25

1 in computer 100. Additional details regarding the verification module 108 and the  
2 digests 110 are provided below.

3 Referring again to Fig. 1, computer 100 is coupled to the Internet 112. A  
4 pair of music servers 114 are also coupled to the Internet 112. The music servers  
5 114 contain various music files stored in the MP3 format. A music file may be an  
6 entire song (also referred to as a “track”), a collection of multiple songs (e.g., the  
7 entire content of a music CD), a portion of a song, or any other type of file  
8 containing audio data. Computer 100 accesses the music servers 114 to download  
9 various music files. The downloaded music files may be installed on handheld  
10 music player 102 for playback by the music player, or may be stored on computer  
11 100 for playback on the computer using a music player application 116. The  
12 music player application 116 is also capable of extracting raw audio data from, for  
13 example, a CD in CD-ROM drive 106 and converting the raw audio data into MP3  
14 music files for playback on personal computer 100 or music player 102. To  
15 discourage unauthorized copying or distribution of MP3 music files, verification  
16 module 108 in music player 102 prevents the installation or playback of music  
17 files unless the MP3 file is legitimately obtained from a music server 114 or a  
18 legitimate copy of the original music CD is present in CD-ROM drive 106. An  
19 exemplary verification procedure is discussed below with respect to Fig. 2.

20 In an alternate embodiment of the invention, music player 102 is coupled  
21 directly to a CD-ROM drive through a communication link. In this embodiment, a  
22 computer is not required because the verification module 108 communicates  
23 directly with the CD-ROM drive to verify that the user of music player 102 has a  
24 legitimate CD that corresponds to the music file to be played or installed on the  
25 music player. In this alternate embodiment, music player 102 may be coupled

1 directly to the Internet 112. This arrangement allows the music player 102 to  
2 download music files across the Internet 112 from music servers 114.

3 Fig. 2 is a flow diagram illustrating a procedure for verifying the existence  
4 of an original data storage medium (for example, a music CD) before playing or  
5 installing a music file. This procedure could be used for before any song is  
6 played, but would more likely be triggered by the existence of a “watermark”  
7 embedded in the song that indicates that the music is protected and warrants  
8 special treatment.

9 Initially, a user requests to install or play a music file (step 120). The  
10 request may be entered at computer 100 or music player 102. Before allowing the  
11 installation or playback of the music file, the verification module 108 randomly  
12 selects a block of data associated with the requested music file (step 122). The  
13 verification module 108 then asks the client (in this case, the computer 100) to  
14 produce the data associated with the selected block from the corresponding music  
15 CD in CD-ROM drive 106 (step 124). After receiving the requested data from  
16 computer 100, the verification module 108 performs a digest operation on the  
17 received data (step 126). The digest operation is a cryptographic operation that  
18 processes a block of data such that the resulting digest is significantly smaller in  
19 size than the original block of data. The requirements for the cryptographic digest  
20 operation are that the resulting digest should be smaller than the original data, but  
21 still large enough to minimize the chances that two data blocks digest to the same  
22 value (typically, a few hundred bits are appropriate), and that it is computationally  
23 infeasible to find two data blocks that digest to the same value. An example of a  
24 suitable hashing operation is SHA (secure hash algorithm).

1        The table of digests 110 shown in Fig. 1 represent a table of the results of  
2        the digest calculation as applied to all blocks of a known legitimate CD. Thus, the  
3        table of digests 110 represent known verification data. The same cryptographic  
4        calculation or algorithm is used in step 126 and to generate the table of digests  
5        110. If the music CD in the CD-ROM drive 106 is legitimate, then the digest of  
6        any block on the CD will match the corresponding digest entry in the table of  
7        digests 110.

8        After performing the digest operation on the received data, the verification  
9        module 108 compares the result of the digest operation to the known verification  
10       data contained in digests 110 (step 128). If the comparison does not result in a  
11       match, then the procedure prevents the installation or playback of the requested  
12       music file (step 132). However, if the comparison results in a match, then the  
13       procedure determines whether to perform additional verification (step 134). In a  
14       particular example, the procedure of Fig. 2 verifies three different random blocks  
15       before determining that the CD in the CD-ROM drive 106 is legitimate. If no  
16       additional verification is required, then the procedure allows the installation or  
17       playback of the requested music file (step 136). If additional verification is  
18       required, then the verification module 108 randomly selects another block of data  
19       (step 138) and returns to request the appropriate block of data from the client (step  
20       124).

21       Fig. 3 illustrates a table of digests 150 for a particular music CD. The first  
22       entry in the table of digests 150 identifies the name of the CD with which the  
23       digest is associated. Each subsequent entry in the table of digests 150 identifies  
24       the result of the digest operation performed on a block of data of a known  
25       legitimate CD. The table of digests 150 is typically created when the CD is

1 created. A copy of the table of digests 150 can be stored on the original CD itself  
2 and may be read from the CD by the verification module. Alternatively, a copy of  
3 the table of digests 150 can be made publicly available on an internet web site. In  
4 the example of Fig. 3, the table of digests 150 contains digests for 100 different  
5 blocks. Thus, the known legitimate CD was partitioned into 100 blocks of data.  
6 The digest operation was performed on each block of data to generate the digests  
7 150. As discussed above, during a verification process, several blocks are selected  
8 at random for verification. If the result of the digest operation on each of the  
9 selected blocks matches the corresponding digest value stored in the table of  
10 digests 150, then the CD in the CD-ROM drive is considered legitimate.

11 If the verification process is performed by a presumed secure web server to  
12 provide additional media or other services to legitimate owners of the original  
13 recording media, a simple table of digests is sufficient. However, in the case of a  
14 device or program in which the verification module is under the control of the  
15 user, additional measures are required to protect the integrity of the table of  
16 digests. Otherwise, a pirate could distribute a fake digest table with the pirated  
17 media. A suitable way of protecting digest lists from tampering is to digitally sign  
18 the digest list with a signature provided by a reputable authority (e.g., a  
19 certification authority). In this case, the verification module can verify that the  
20 digest table comes from a know authority by checking its certificate. Additionally,  
21 the verification module can check that the digest list has not been tampered by  
22 checking that the signature matches the data in the table. Many digital signature  
23 methods are available and appropriate. An suitable example is DSA, or the digital  
24 signature algorithm.

25

1        The use of digests discussed above significantly reduces the amount of data  
2        that must be stored by the verifying device. For example, music player 102 in Fig.  
3        1 stores the table of digests 110 which is significantly smaller than the entire  
4        content of the CD in drive 106. Additionally, by randomly selecting a few data  
5        blocks for verification, the amount of data communicated between the music  
6        player 102 and the personal computer is significantly reduced. Since the selected  
7        data blocks are chosen at random, a person trying to distribute pirated copies  
8        cannot know which data blocks will be selected during a particular verification  
9        process.

10       Fig. 4 illustrates an exemplary computer system 200 containing two  
11       application programs 206 and 210 and a verification module 204. Computer  
12       system 200 also contains a CD-ROM drive 202. When attempting to launch either  
13       the game application program 206 or the calendar application program 210, the  
14       verification module 204 verifies that CD-ROM drive 202 contains an original  
15       program CD containing the application being launched. This verification helps  
16       discourage software piracy by requiring the presence of an original program CD  
17       prior to launching the application program. Each application program 206 and  
18       210 has embedded therein a table of digests 208 and 212, respectively.  
19       Verification module 204 uses the data contained in digests 208 and 212 during the  
20       verification process.

21       In this case, the verification module is embedded in the application itself,  
22       which would allow a pirate to disable the verification part of the application  
23       installation and launch to defeat the verification module. To make disabling the  
24       verifier harder, a software publisher could use the techniques of "software tamper  
25       resistance" which makes it difficult for an attacker to modify a program without

1 the program refusing to run, or running improperly. An example of suitable  
2 techniques is described in "Tamper Resistant Software: An Implementation",  
3 David Aucsmith, IHW'96 - Proc. of the First International Information hiding  
4 Workshop, Vol. 1174 (1997), pp. 317-333.

5 The table of digests can also be protected using the signature technique  
6 already described, or be embedded into the application itself in a way that is hard  
7 to modify.

8 Fig. 5 is a flow diagram illustrating a procedure for verifying the existence  
9 of an original compact disc before launching an application program. The  
10 procedure begins when a user requests to launch an application program (step  
11 220). The verification module randomly selects a block of data to be verified (step  
12 222). The verification module reads the data associated with the selected block  
13 from the program CD in CD-ROM drive 202 (step 224) and performs a digest  
14 operation on the data read from the program CD (step 226). The result of the  
15 digest operation is then compared to the known verification data contained in the  
16 table of digests associated with the application to be launched (step 228). If the  
17 result of the digest operation does not match the known verification data, then the  
18 procedure does not launch the requested application (step 232). In this situation, a  
19 message may be displayed to the user of the computer system requesting the  
20 insertion of the original program CD into the CD-ROM drive.

21 If the result of the digest operation matches the known verification data,  
22 then the procedure determines whether additional verification is necessary (step  
23 234). If no additional verification is necessary, then the procedure launches the  
24 requested application (step 236). If additional verification is necessary, then the  
25

1 verification module selects another block of data (step 238) and returns to read the  
2 data associated with the selected block from the program CD (step 224).

3 As discussed above, many application programs do not utilize all of the  
4 storage space available on a CD. In these instances, the unused portions of the CD  
5 (which would otherwise be empty) can be filled with random data. The entire CD  
6 (including the random data) is partitioned into blocks, thereby discouraging the  
7 production of pirated CDs that contain multiple applications.

8 Fig. 6 illustrates an exemplary environment in which a remote server  
9 verifies whether an original CD is present in a local computer system 300.  
10 Computer system 300 is coupled to servers 302 and 304 via the Internet 306. The  
11 computer system 300 includes an application program 308 and a CD-ROM drive  
12 310. Server 302 includes a verification module 312 and a table of digests 314.  
13 Similarly, server 304 includes a verification module 316 and a table of digests 318.  
14 When a user of computer system 300 attempts to launch application program 308,  
15 the application program initiates contact with an appropriate server, such as the  
16 application program manufacturer's web server. Once contacted, the server's  
17 verification module verifies that the original program CD is in the CD-ROM drive  
18 310. The server accomplishes this verification by requesting (through its  
19 verification module) that the computer system provide certain blocks of data from  
20 the program CD in the CD-ROM drive 310. The requested data is read from the  
21 program CD and communicated across the Internet 306 to the verification module.  
22 The verification module performs a digest operation on the received data and  
23 compares the result to the corresponding entry in the table of digests. If all of the  
24 data blocks match, then the server sends authorization to the computer system 300  
25

1 to launch the application program. If one of the data blocks does not match, then  
2 the server instructs the computer system 300 not to launch the application  
3 program.

4 The configuration shown in Fig. 6 may require the transmission of a  
5 significant amount of data between computer system 300 and the server 302 or  
6 304 during the verification process. For example, if the program CD is partitioned  
7 into 100 blocks, each block may contain over 5 Mb of data. If the verification  
8 process reads four blocks of data, 20 Mb of data would be transmitted across the  
9 Internet 306. Increasing the number of blocks will reduce the amount of data that  
10 must be transmitted across the Internet for each block. For example, increasing  
11 the number of blocks to 1000 reduces the amount of data per block to  
12 approximately 500 Kb. If the verification process reads four blocks of data, 2 Mb  
13 of data would be transmitted across the Internet 306.

14 The amount of data transmitted across the Internet 306 during the  
15 verification process can be further reduced using a keyed-hash or message  
16 authentication code (MAC) function. In this situation, the verification module in  
17 the server provides a randomly selected data block number and a challenge  
18 (typically, a random number) to the computer system 300. The computer system  
19 300 hashes together the challenge and the content of the selected data block on the  
20 program CD. A suitable message authentication code is Message Authentication  
21 Algorithm (MAA). The computer system 300 then returns the result of the hash  
22 operation to the server. This procedure significantly reduces the amount of data  
23 that is transmitted across the Internet 306 because the hash operation is performed  
24 on the computer system 300, thereby eliminating the need to transmit the block  
25 data across the Internet. Software pirates cannot predict the result of the hash

1 operation because both the challenge and the data block are selected at random.  
2 Although this use of a challenge and a hash operation has been described with  
3 reference to the embodiment of Fig. 6, this procedure can be used in any of the  
4 embodiments discussed above.

5 A particular embodiment of the invention may be used by a manufacturer to  
6 distribute "bonus" music tracks to purchasers of music CDs. For example, a  
7 customer purchases a particular music CD. The manufacturer of the music CD  
8 offers free additional music tracks in MP3 format, available from the  
9 manufacturer's music server (also referred to as a web site). Before the customer  
10 is permitted to download the additional music tracks, the manufacturer verifies  
11 that the customer has the original music CD in the CD-ROM drive. In this  
12 situation, the verification module is located in the manufacturer's music server,  
13 and the random data blocks are retrieved from the music CD across the Internet. If  
14 an original music CD is verified, then the music server downloads the bonus music  
15 tracks to the customer for playback on a personal computer or a handheld music  
16 player.

17 In another exemplary use of the invention, purchasers of application  
18 programs may download upgrades or "bonus" material related to the application  
19 program. The manufacturer verifies that the customer has an original program CD  
20 using the procedures discussed above. If an original program CD is verified, then  
21 the manufacturer downloads an application upgrade and/or additional materials to  
22 the customer. If an original program CD cannot be verified, then the upgrade and  
23 additional materials are not downloaded.

24 Fig. 7 shows a general example of a computer 430 that can be used with the  
25 present invention. A computer such as that shown in Fig. 7 can be used, for

1 example, to perform various procedures necessary to verify that an original CD is  
2 present in the CD-ROM drive, and to run various applications, such as a music  
3 player application. The computer shown in Fig. 7 can also be used to perform the  
4 calculations necessary to compute the digest value associated with particular  
5 blocks of data. Furthermore, the computer shown in Fig. 7 can function as a  
6 server (such as a music server) of the type discussed above.

7 Computer 430 includes one or more processors or processing units 432, a  
8 system memory 434, and a bus 436 that couples various system components  
9 including the system memory 434 to processors 432. The bus 436 represents one  
10 or more of any of several types of bus structures, including a memory bus or  
11 memory controller, a peripheral bus, an accelerated graphics port, and a processor  
12 or local bus using any of a variety of bus architectures. The system memory 434  
13 includes read only memory (ROM) 438 and random access memory (RAM) 440.  
14 A basic input/output system (BIOS) 442, containing the basic routines that help to  
15 transfer information between elements within computer 430, such as during start-  
16 up, is stored in ROM 438.

17 Computer 430 further includes a hard disk drive 444 for reading from and  
18 writing to a hard disk (not shown), a magnetic disk drive 446 for reading from and  
19 writing to a removable magnetic disk 448, and an optical disk drive 450 for  
20 reading from or writing to a removable optical disk 452 such as a CD ROM or  
21 other optical media. The hard disk drive 444, magnetic disk drive 446, and optical  
22 disk drive 450 are connected to the bus 436 by an SCSI interface 454 or some  
23 other appropriate interface. The drives and their associated computer-readable  
24 media provide nonvolatile storage of computer-readable instructions, data  
25 structures, program modules and other data for computer 430. Although the

1 exemplary environment described herein employs a hard disk, a removable  
2 magnetic disk 448 and a removable optical disk 452, it should be appreciated by  
3 those skilled in the art that other types of computer-readable media which can  
4 store data that is accessible by a computer, such as magnetic cassettes, flash  
5 memory cards, digital video disks, random access memories (RAMs), read only  
6 memories (ROMs), and the like, may also be used in the exemplary operating  
7 environment.

8 A number of program modules may be stored on the hard disk 444,  
9 magnetic disk 448, optical disk 452, ROM 438, or RAM 440, including an  
10 operating system 458, one or more application programs 460, other program  
11 modules 462, and program data 464. A user may enter commands and  
12 information into computer 430 through input devices such as a keyboard 466 and a  
13 pointing device 468. Other input devices (not shown) may include a microphone,  
14 joystick, game pad, satellite dish, scanner, or the like. These and other input  
15 devices are connected to the processing unit 432 through an interface 470 that is  
16 coupled to the bus 436. A monitor 472 or other type of display device is also  
17 connected to the bus 436 via an interface, such as a video adapter 474. In addition  
18 to the monitor, personal computers typically include other peripheral output  
19 devices (not shown) such as speakers and printers.

20 Computer 430 commonly operates in a networked environment using  
21 logical connections to one or more remote computers, such as a remote computer  
22 476. The remote computer 476 may be another personal computer, a server, a  
23 router, a network PC, a peer device or other common network node, and typically  
24 includes many or all of the elements described above relative to computer 430,  
25 although only a memory storage device 478 has been illustrated in Fig. 7. The

1 logical connections depicted in Fig. 7 include a local area network (LAN) 480 and  
2 a wide area network (WAN) 482. Such networking environments are  
3 commonplace in offices, enterprise-wide computer networks, intranets, and the  
4 Internet.

5 When used in a LAN networking environment, computer 430 is connected  
6 to the local network 480 through a network interface or adapter 484. When used  
7 in a WAN networking environment, computer 430 typically includes a modem 486  
8 or other means for establishing communications over the wide area network 482,  
9 such as the Internet. The modem 486, which may be internal or external, is  
10 connected to the bus 436 via a serial port interface 456. In a networked  
11 environment, program modules depicted relative to the personal computer 430, or  
12 portions thereof, may be stored in the remote memory storage device. It will be  
13 appreciated that the network connections shown are exemplary and other means of  
14 establishing a communications link between the computers may be used.

15 Generally, the data processors of computer 430 are programmed by means  
16 of instructions stored at different times in the various computer-readable storage  
17 media of the computer. Programs and operating systems are typically distributed,  
18 for example, on floppy disks or CD-ROMs. From there, they are installed or  
19 loaded into the secondary memory of a computer. At execution, they are loaded at  
20 least partially into the computer's primary electronic memory. The invention  
21 described herein includes these and other various types of computer-readable  
22 storage media when such media contain instructions or programs for implementing  
23 the steps described below in conjunction with a microprocessor or other data  
24 processor. The invention also includes the computer itself when programmed  
25 according to the methods and techniques described herein.

1 For purposes of illustration, programs and other executable program  
2 components such as the operating system are illustrated herein as discrete blocks,  
3 although it is recognized that such programs and components reside at various  
4 times in different storage components of the computer, and are executed by the  
5 data processor(s) of the computer.

6 Alternatively, the invention can be implemented in hardware or a  
7 combination of hardware, software, and/or firmware. For example, one or more  
8 application specific integrated circuits (ASICs) could be programmed to carry out  
9 the invention.

10 Thus, a system has been described that verifies the existence of an original  
11 data storage medium, such as a CD or DVD, without requiring an analysis of  
12 every byte of data stored on the CD or DVD. Although the invention has been  
13 described in language specific to structural features and/or methodological steps, it  
14 is to be understood that the invention defined in the appended claims is not  
15 necessarily limited to the specific features or steps described. Rather, the specific  
16 features and steps are disclosed as preferred forms of implementing the claimed  
17 invention.

18

19

20

21

22

23

24

25